**Overview**

The TFN Project is divided into two phases, focusing on data ingestion, processing, and web-based visualization.

**Phase 1: Data Ingestion & Processing**

* A Python script is developed to fetch data from an API.
* The script cleans, processes, and inserts the data into a PostgreSQL database.
* Certain predefined or hardcoded data is stored in a structured JSON file, which is used for mapping with the database entries.

**Phase 2: Web-Based Visualization**

* An in-house website, similar to a contact center, is developed to fetch and display data from the PostgreSQL database.
* This website enables users to interact with the processed data efficiently.

**Enhancements in Phase 1: CI/CD and Distributed Scheduling**

**CI/CD Implementation with Azure DevOps (ADO)**

* The Phase 1 Python script will be integrated into a CI/CD pipeline in Azure DevOps.
* The pipeline automates the deployment and execution of the script, ensuring seamless data updates.

**Scheduled Execution from Different Servers**

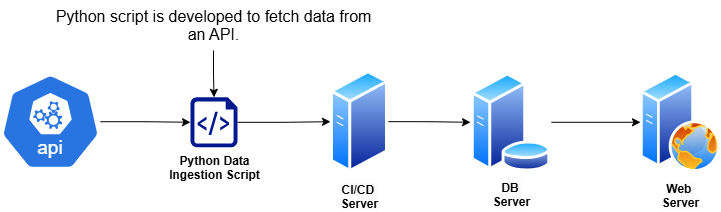
* Instead of scheduling the script on the same server hosting the website, it will be scheduled to run every hour on a dedicated processing server.

**Reasons for Running the Script on a Separate Server**

1. **Performance Optimization:** Running the data ingestion script on the same server as the website could impact the website’s performance due to resource consumption.
2. **Load Distribution:** Dedicated servers for data ingestion prevent CPU and memory-intensive operations from affecting website responsiveness.
3. **Scalability & Reliability:** Using different servers allows horizontal scaling, reducing the risk of a single point of failure.
4. **Security & Isolation:** Keeping database operations separate from the web server enhances security and reduces exposure to potential vulnerabilities.

**High-Level Architecture Diagram**

**Deployment Strategy**



* The CI/CD pipeline ensures that the Python script is automatically deployed and executed at scheduled intervals.
* Three separate servers are used:
  + One for the PostgreSQL database.
  + One for executing the data ingestion script.
  + One for hosting the web application.
* The web application queries the PostgreSQL database for real-time data display.